REMARKS

Applicants respectfully request consideration of the subject application. Applicants amended claims 1, 6, 11, 16, 21, 25, and 30 to clarify the limitations already present in the claims, thus Applicants added no new matter through the amendments.

Claim Rejections Under 35 U.S.C. § 102(e)

Claims 16-18, 20-22 and 24 have been rejected under 35 U.S.C. § 102(e) as being anticipated by Blaker, et al. U.S. Publication No. 2002/0004904 ("Blaker").

Claim 16

Applicants respectfully disagree with the rejection because Blaker fails to describe each and every element of claim 16. Applicants request reconsideration of the rejected claim.

Claim 16 requires a first processor to call a macro security operation associated with a *plurality* of primitive security operations. Moreover, claim 16 further requires a second processor to perform the *plurality* of primitive security operations in response to the macro security operation from said first processor.

Blaker describes a host processor that loads a command block into a command queue. (Blaker, para. [0039]). The cryptographic accelerator processor accesses the command queue, downloads the command block, and executes the command block. (Blaker, para. [0039]). Blaker describes loading a command block into memory so "the host processor 16 need not spend time interacting directly with

the cryptographic accelerator processor 14 (e.g., issuing a command to the cryptographic accelerator processor 14, waiting for that command to complete, and then issuing another command)." (Blaker, para. [0039]). The command block as illustrated in figures 12A-12B and Figures 14A, 14B, and 15 provides an operation to be executed by a cryptographic accelerator processor. (Blaker, para. [0039]). The command block as described by Blaker may also include an interrupt field (Blaker, para. [0049] and Figure 12A), a location to store information (Blaker, paras. [0049] and [0054], and Figure 12B and Figure 14B), a location to update a completion field (Blaker, para. [0050] and Figure 12C), or a location to store results (Blaker, para. [0050] Figure 12D). Blaker further describes the use of a command block that includes parameters such as a hash key and other input information needed for the cryptographic accelerator to complete an instruction (Blaker, para. [0054] and [0055] and Figure 14A and Figure 15).

Blaker describes a one to one relationship between operand to instruction. For instance, paragraph 11 describes, "one or more operands are downloaded into the local memory from the system memory and the cryptographic processor executes *an instruction* that references *one* of the downloaded operands." (Blaker, para. 11, emphasis added). Thus, for every operand in the command block the cryptographic accelerator processor executes one instruction. Therefore, Blaker fails to describe a *macro* security operation associated with a *plurality* of primitive security operations.

Furthermore, claim 16 achieves a different result than Blaker. In claim 16, the result of having a first processor to call a macro security operation associated to a

number of primitive security operations minimizes the number of instructions that the first processor must send to the second processor. Conversely, the goal of Blaker is to achieve the result of eliminating the need of the host processor to interact directly with the cryptographic accelerator processor. (Blaker, para. [0039]). Furthermore, Blaker describes the size and number of command block sequences as being less constrained because the availability of system memory is generally more abundant. (Blaker, para. [0039]). Because the size and number of command block sequences is less constrained the host processor can increase the size and number of sequences sent to the cryptographic accelerator processor, which is contrary to the claim 16 outcome of minimizing instructions sent to a second processor.

Because Blaker fails to describe a first processor to call a *macro* security operation associated with a *plurality* of primitive security operations and a second processor to perform a *plurality* of primitive security operations in response to the *macro* security operation from said first processor, Blaker fails to render claim 16 obvious.

Claim 17, 18, and 20

Applicants respectfully submit that claims 17, 18, and 20 are dependent on claim 16; therefore, claims 17, 18, and 20 include the same limitations as claim 16. As such, claims 17, 18, and 20 are patentable for at least the same reasons as claim 16.

Claim 21

Applicants respectfully disagree with the rejection because Blaker fails to describe each and every element of claim 21. Applicants request reconsideration of the rejected claim.

Claim 21 requires a first processor to give the command for a macro security operation associated with a *plurality* of primitive security operations. Moreover, claim 21 requires a request unit to retrieve the macro security operation associated with the *plurality* of primitive security operations. Another requirement of claim 21 includes one of a plurality of execution units to perform the *plurality* of primitive security operations retrieved by the request unit, the *plurality* of primitive security operations corresponding to *the macro security operation*.

As discussed above, Blaker describes a command block created by a host processor. The command block includes commands for execution, locations for storing, and parameters to be loaded by the cryptographic accelerator processor. Blaker describes a one to one relationship between operand to instruction. For instance, paragraph 11 describes, "one or more operands are downloaded into the local memory from the system memory and the cryptographic processor executes *an instruction* that references *one* of the downloaded operands." (Blaker, para. 11, emphasis added). Thus, for every operand in the command block the cryptographic accelerator processor executes one instruction. Therefore, Blaker fails to describe a *macro* security operation associated with a *plurality* of primitive security operations.

Because Blaker fails to describe a first processor to give the command for a macro security operation associated with a *plurality* of primitive security operations,

a request unit to retrieve the macro security operation associated with the *plurality* of primitive security operations, and one of a plurality of execution units to perform the *plurality* of primitive security operations *corresponding* to the macro security operation, Blaker fails to render claim 21 obvious.

Claim 22

Applicants respectfully submit that claim 22 is dependent on claim 21; therefore, claim 22 includes the same limitations as claim 21. As such, claim 22 is patentable for at least the same reasons as claim 21.

Claim Rejections Under 35 U.S.C. §103(a)

Claims 1-15 and 25-34 have been rejected under 35 U.S.C. §103(a) as being unpatentable over Blaker in view of Bellwood et al., U.S Patent No. 6,584,567 ("Bellwood"). Claims 19 and 23 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over Blaker further in view of Tremblay et al., U.S. Patent No. 5,925,123 ("Tremblay").

Claim 1

Claim 1 requires calling with a *single* macro instruction operation from a first processor, the *single* macro instruction operation representing a *plurality* of primitive security operations. Moreover, claim 1 requires executing the *plurality* of primitive security operations at a second processor in response to receiving the *single* macro

instruction operation from the first processor.

As discussed above, Blaker describes a command block created by a host processor. The command block includes commands for execution, locations for storing, and parameters to be loaded by the cryptographic accelerator processor. Bellwood describes a method of enabling a proxy to participate in a secure communication between a client and a set of servers. (Bellwood, Abstract). A first secure connection is established between the client and the proxy. (Bellwood, Abstract). The method also describes the proxy participating in secure communications between the client and a first server. (Bellwood, Abstract). Because the client, the proxy, and the first server must participate in secure communications, the three must have cryptographic functionality. Therefore, the combination of Blaker and Bellwood describe a client, a proxy, and a set of servers each having a host processor and a cryptographic accelerator processor, where the host processor creates command blocks to be loaded by the cryptographic accelerator processor as described by Blaker.

Blaker describes a one to one relationship between operand to instruction. For instance, paragraph 11 describes, "one or more operands are downloaded into the local memory from the system memory and the cryptographic processor executes *an instruction* that references *one* of the downloaded operands." (Blaker, para. 11, emphasis added). Thus, for every operand in the command block the cryptographic accelerator processor executes one instruction. Conversely, claim 1 requires the use of *an operation* to represent a *plurality* of primitive security operations. An example but not a limitation of an operation representing a plurality

of primitive security operations is illustrated in Applicants' Figure 5. Another example but not a limitation of an operation representing a plurality of primitive security operations according to the embodiment of claim 1 is disclosed in the Specification and illustrated in Figure 7. The example demonstrates the use of one operation sent to a second processor which then executes 39 primitive security operations. (Application, paras. [0043]-[0045] and Figure 7).

Furthermore, claim 1 achieves a different result than Blaker. In claim 1, the result of calling with a single macro instruction representing a plurality of primitive security operations, according to claim 1, minimizes the number of instructions that the first processor must send to the second processor. Conversely, the goal of Blaker is to achieve the result of eliminating the need of the host processor to interact directly with the cryptographic accelerator processor. (Blaker, para. [0039]). Furthermore, Blaker describes the size and number of command block sequences as being less constrained because the availability of system memory is generally more abundant. (Blaker, para. [0039]). Because the size and number of command block sequences is less constrained the host processor can increase the size and number of sequences sent to the cryptographic accelerator processor, which is contrary to the claim 1 outcome of minimizing instructions sent to a second processor.

Because the combination of Blaker and Bellwood fails to describe calling with a *single* macro instruction operation from a first processor, the *single* macro instruction operation representing a *plurality* of primitive security operations and executing the *plurality* of primitive security operations at a second processor in

response to receiving *the single macro instruction operation* from the first processor, the combination fails to render claim 1 obvious.

Claims 2-5

Applicants respectfully submit that claims 2-5 are dependent on claim 1; therefore, claims 2-5 include the same limitations as claim 1. As such, claims 2-5 are patentable for at least the same reasons as claim 1.

Claim 6

Claim 6 requires similar limitations as claim 1. Specifically, claim 6 requires calling *a macro security operation* from a first processor to a second processor, the macro security operation representing *a set of operations* and performing the *set of operations* in response to *the macro security operation*. As discussed above, the combination of Blaker and Bellwood fails to describe a macro security operation representing a set of operations; therefore, the combination fails to render claim 6 obvious for at least the reasons discussed for claim 1.

Claims 7-10

Applicants respectfully submit that claims 7-10 are dependent on claim 6; therefore, claims 7-10 include the same limitations as claim 6. As such, claims 7-10 are patentable for at least the same reasons as claim 6.

Claim 11

Claim 11 requires similar limitations as claim 1. Specifically, claim 11 requires the second network element to call a macro security operation from a first processor, *the macro security operation* associated with *a plurality of primitive security operations* and to execute *the plurality of primitive security operations* at a second processor in response to *the macro security operation*. As discussed above, the combination of Blaker and Bellwood fails to describe *a macro security operation* associated with *a plurality of primitive security operations*; therefore, the combination fails to render claim 11 obvious for at least the reasons discussed for claim 1.

Claims 12-15

Applicants respectfully submit that claims 12-15 are dependent on claim 11; therefore, claims 12-15 include the same limitations as claim 11. As such, claims 12-15 are patentable for at least the same reasons as claim 11.

Claims 25

Claim 25 requires similar limitations as claim 1. Specifically, claim 25 requires executing a macro security operation at a first one of the set of processors, *the macro security operation* associated with a *plurality* of primitive security operations and executing a *plurality* of primitive security operations at a second one of the set of processors in response to *the macro security operation*. As discussed above, the combination of Blaker and Bellwood fails to describe *a macro security operation* associated with a *plurality* of primitive security operations; therefore, the

combination fails to render claim 25 obvious for at least the reasons discussed for claim 1.

Claims 26-29

Applicants respectfully submit that claims 26-29 are dependent on claim 25; therefore, claims 26-29 include the same limitations as claim 25. As such, claims 26-29 are patentable for at least the same reasons as claim 25.

Claim 30

Claim 30 requires similar limitations as claim 1. Specifically, claim 30 requires calling a macro security operation from a first one of the set of processors, *the macro security operation* associated with *a set of operations* and performing *the set of operations* at a second one of the set of processors in response to *the macro security operation*. As discussed above, the combination of Blaker and Bellwood fails to describe *a macro security operation* associated with *a set of operations*; therefore, the combination fails to render claim 25 obvious for at least the reasons discussed for claim 1.

Claims 31-34

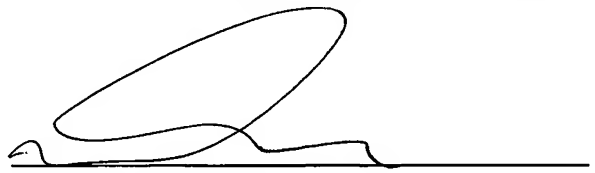
Applicants respectfully submit that claims 31-34 are dependent on claim 30; therefore, claims 31-34 include the same limitations as claim 30. As such, claims 31-34 are patentable for at least the same reasons as claim 30.

Conclusion

If the allowance of these claims could be facilitated by a telephone conference, the Examiner is invited to contact the undersigned at (408) 720-8300. If there are any additional charges, please charge our Deposit Account No. 02–2666.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP



Dated: 1/23, 2006

Daniel M. De Vos
Registration No. 37,813

Customer No. 08791
12400 Wilshire Boulevard
Seventh Floor
Los Angeles, CA 90025-1030
(408) 720-8300